MicroConverter®, Multi-Channel 12-bit ADC with Embedded FLASH MCU

Errata Sheet — ADuC812

A. This Errata sheet represents the known bugs, anomalies and work-arounds for the ADuC812 MicroConverter.

B. The Errata listed, apply to all ADuC812 packaged material branded as follows :

(Production Material)
ADUC812BS
Date Stamp   AD Logo
Lot Number

Where the Date Stamp number is greater than or equal to 9933.

(**Engineering Lot Exception** : Errata Listed in this sheet also apply to parts branded with Lot Number/Date Stamp combination = F104366.1 / 9925.)

C. Errata Sheets referring to previous silicon revisions are available on our web-site at
http://www.analog.com/microconverter/erratasheets.html

D. Because of silicon revisions and improved work-arounds, this Errata Sheet will be updated periodically. Please find the latest Errata Sheet listed on the World Wide Web at :
http://www.analog.com/microconverter/erratasheets.html

E. Analog Devices Inc. is committed, through future silicon revisions to continuously improve silicon functionality. Analog Devices Inc. will use its best endeavors to ensure that these future silicon revisions remain compatible with your present software/systems that implement the recommended work-arounds outlined in this document.

REV. E.2   Sept/01

1. ADC FUNCTION - ADC CONVERSION IN DMA MODE :

Background: The ADC can be triggered by 1 of 4 trigger events in Normal and DMA operating modes, namely: Single Software Conversion, Continuous Software Conversions, External Pin Triggered Conversions and Timer2 triggered Conversions. DMA mode is initiated by setting the DMA bit in the ADCCON2 SFR and the DMA controller proceeds starting from the external data memory address indicated by the DMA address SFRs DMAL/H/P. Once a DMA conversion sequence is complete, both the DMA bit and the bit controlling the conversion mode are reset by the DMA controller.

Issue: a. DMA operation will not execute when triggered from Timer2 driven conversions unless DMA Address (DMAL/H/P) is initially set to 000000H.

b. The conversion mode bits (EXTC and T2C) are NOT reset upon completion of a DMA conversion sequence triggered by Timer2 or external CONVST events.

c. A write to DMA address registers DMAL or DMAH can corrupt the LSB of the adjacent DMA address register byte (DMAH or DMAP).

d. DMA operation will not execute when triggered from Timer2 driven conversions unless ADCCON2 is written before ADCCON1 when configuring the ADC.

Work-Around : a. For Timer2 triggered DMA conversions, ensure that the DMA start address is set to 000000H.

b. When using Timer2 or Ext. Pin triggered DMA conversions, ensure that you force EXTC and T2C bits (ADCCON2) to reset by writing to these bits immediately after DMA completion. This will ensure that the ADC halts conversion as soon as the DMA sequence has completed.

c. The DMA address registers should be written in the following order: DML, DMAH, DMAP.

d. For Timer2 triggered conversions, ensure the correct configuration sequence is used for the ADC i.e. Configure ADCCON2 before ADCCON1.

Related Issues : None.

2. ADC FUNCTION - TIME BETWEEN ADC CONVERSIONS

Background: The ADC can be triggered by 1 of 4 trigger events in Normal and DMA operating modes, namely: Single Software Conversion, Continuous Software Conversions, External Pin Triggered Conversions and Timer2 triggered Conversions. In all modes of operation the ADC is specified for 12-bit INL and DNL performance.

Issue: If an ADC conversion is initiated 100msecs or more after the last conversion has taken place, the accuracy of this new conversion result cannot be guaranteed.

Work-Around: Ignore or repeat the first conversion if the time between this conversion and the previous conversion is greater than 100msecs.

Related Issues: None

---

3. UART FUNCTION - MULTIPROCESSOR COMMUNICATIONS MODE :

Background: UART Modes 2 and 3 (programmed via SCON) have a special provision for multiprocessor communications. In these modes, nine data bits are received and the ninth bit goes into RB8. In this mode the port can be programmed so that when the stop bit is received, the serial port interrupt is activated only if RB8(SCON) = 1. This feature is enabled by setting the SM2 bit in SCON.

Issue: With SM2 set, the serial port interrupt is set each time a stop bit is received, regardless of the state of the RB8 bit.

Work-Around :   Implement a software polling scheme in the Serial Port ISR so that the state of the RB8 bit is checked on entering the ISR. If RB8 is '0' then a 'RETI' is executed to exit the ISR immediately, and only if RB8 is set will the ISR function be executed.

Related Issues :   None.

---

4. SPI FUNCTION - OPERATION OF THE SPI INTERRUPT BIT ISPI :

Background: The SPI Interrupt bit ISPI in the SPI Control SFR (SPICON) is set at the end of an SPI transfer (8 SPI Clocks). Typically this bit can be used in an interrupt driven routine to vector to the SPI Interrupt Vector Address at 003BH. Alternatively this bit can be polled in software to detect a new reception or completion of a byte transmission.

Issue: In an interrupt driven routine this bit is automatically cleared when the PC vectors to the SPI ISR. However, in a polled implementation the bit is not writable and is not cleared on subsequent accesses to the SPI data SFRs. This means that in a software polled scheme the ISPI bit is set after the first transmission but will not and cannot be cleared for subsequent byte communications which can result in subsequent reads of the same initial byte or corruption of a byte transmission before it is completed.

Work-Around :   a. Use the SPI peripheral in interrupt driven mode only, thereby avoiding any issues with the ISPI bit as this bit is automatically set and cleared in an SPI interrupt driven routine.

REV. E.2   Sept/01

Work-Around :   b. In master mode SPI operation, use dummy 'NOP' instructions to time out a complete SPI 8-Clock cycle before trying to access the SPI data SFRs.

Related Issues :   None.

---

5. SPI FUNCTION - SLAVE MODE FUNCTIONALITY :

Background: In synchronous slave mode (CPOL=0/1, CPHA=0) the new MSB output value appears on MISO when $\overline{SS}$ goes low.

Issue: The SPI slave does not function correctly as a transmitter in modes 00 and 10 (i.e. when CPHA=0). The MSB of the transmitted byte is corrupted by the MSB of the byte just received.

Work-Around :   To transmit in Slave mode use the alternate SPI Slave operating mode (CPOL=0/1, CPHA=1).

Related Issues :   None.

---

6. PORT 2 AND PORT 3.3 INTERNAL ACTIVE PULL-UP FUNCTIONALITY :

Background :   As with standard 8051 compatible Port functionality, Port 2 and Port 3 pins feature weak internal pull-up resistors and will source current when pulled low externally. On a 0-to-1 transition, an additional pull-up is turned on for 2 master clock periods to help increase transition speed.

Issue :   a. On Port 2 pins, the additional pull-up resistor does not come on during the required 2 clock periods when these pins are being driven as digital outputs.
     b. Port 3.3 is configured as an open drain output without any internal pull-up resistors.

Work-Around :   External pull-up resistors must be used to pull up any Port 2 or P3.3 pin being used as an digital output to ensure fast edges on 0-to-1 transitions.

Related Issues :   Port 2 pins operate correctly (including internal pull-up resistors) when operating as part of the external memory interface on the device.

---

7. FLASH/EE MEMORY - 3V PROGRAMMING :

Background :   The ADuC812 contains 2 separate arrays of Flash/EE memory, namely :

- 8 KBytes Internal Code Space .
- 640 Bytes Flash/EE Data Memory.

The 8 KBytes code space can be programmed in serial download or parallel programming modes, while the 640 Byte data space can be written by user software via a specified SFR interface.

Issue :   This revision of silicon does not support programming the 8K code or 640 Byte data Flash/EE memories at DV_{DD} power supplies of less than 4.5V.

Work-Around :   The device should be programmed at DV_{DD}=5V.

Related Issues :   None.

8. CODE EXECUTION - TRANSITION FROM INTERNAL TO EXTERNAL CODE SPACE :

Background : As with standard 8051 microcontroller functionality the ADuC812 will execute from internal code space when the $\overline{\text{EA}}$ pin is tied high. If the PC rolls over the 8K internal code boundary or is vectored to a location greater than 8K (using a jump instruction for example), the external program memory interface is enabled so as to continue code execution in the external code space.

Issue : A transition from internal to external code will not function correctly. Specifically the P0 outputs designating the low address byte during an op-code fetch is not driven correctly during the first op-code byte fetch cycle, resulting in incorrect or invalid op-code execution. This anomaly will occur during a code rollover to external memory or a 'jmp' or 'call' to a subroutine residing in external memory.

Work-Around : User software should execute a 'MOV P0,#00H' instruction immediately before a rollover, jump or call to external code space is executed to avoid this anomaly.

Related Issues : None.

9. FUNCTIONALITY OF PSMCON.1 BIT IN THE PSMCON SFR :

Background : The PSMCON.1 bit in the PSMCON SFR can be used to indicate which supply caused the active power supply trip condition. PSMCON.1=0 indicates a DV_{DD} fault while PSMCON.1=1 indicates an AV_{DD} fault.

Issue : If the power supply trip point selection bits are set to select the 4.63V or 4.37V supply trip points, the PSMCON.1 (AV_{DD}/DV_{DD} power supply trip indicator) bit may not correctly indicate which supply has caused the fault. The supply trip indicator bit will function correctly at all other trip point voltage levels.

Work-Around : Pending.

Related Issues : It should be noted that while the trip indicator bit function is suspect at these trip levels, the PSM function itself is still 100% operational (even at 4.63V and 4.37V trip levels) generating an interrupt as soon as any one of the supply pins falls below the specified supply voltage trip point selection.

10. FLASH MEMORY - SECURITY BITS FUNCTIONALITY :

Background: 3 undocumented security bits exist at Page 160 (A0H) of User Flash/EE memory. The security bits can be accessed in parallel programming mode and through the SFR read/write interface to the User Flash/EE memory.

Issue: The security bits do not function correctly. Do NOT program page 160 (A0H) of User Flash/EE memory.

Work-Around : Pending.

Related Issues : None.

11. XTAL1 LOGIC INPUT VOLTAGE :

The XTAL1 input is NOT included in the datasheet Input High Voltage Specification. The input high voltage specification for this pin is specified as 4.0V minimum with DV_{DD}=5V.

12. ENTRY TO SERIAL DOWNLOAD MODE:

Background: If the $\overline{PSEN}$ pin (#41) is pulled low via an external pull-down resistor during a power cycle or during an assertion and de-assertion of the external RESET pin(#15), the ADuC812 will enter serial download mode.

Issue: Some ADuC812 devices, when subjected to the above conditions may fail to enter serial download mode and will instead enter normal operating mode.

Work-Around :   a. For devices with kernel revision[1] previous to V210 :
- Parallel program any device that fails to enter serial download mode.

b. For devices with kernel revision[1] V210 or greater :
(i) These devices will correctly enter serial download mode at or below temperatures of 25°C.
(ii) For serial programming at elevated temperatures, a subsequent re-toggle of the RESET pin with the $\overline{PSEN}$ pin pulled low will guarantee entry into serial download mode.

[1] The kernel revision is denoted in the serial ID character string returned via the ADuC812 serial port when the part is configured into serial download mode. Refer to Technical Note: uC004 at **http://www.analog.com/microconverter** for more information on serial download mode.

Related Issues :   a. When executing code from external memory, failure to correctly readback bytes from the 640 Flash/EE data space is also a symptom of this errata.
b. Refer to Tools Change Notification, 812tcn02.pdf available at **http://www.analog.com/microconverter/download/812tcn02.pdf** for further details on kernel revision V210.

13. EXTERNAL POWER-ON RESET (POR) OPERATION - DOCUMENTATION UPDATE :

All ADuC812 documentation has been updated to include a detailed recommendation on how the external RESET pin (#15) should be driven with external POR circuitry during power-on, power-down and brown-out conditions. These recommendations can be found in the ADuC812 technical datasheet (REVA) and the ADuC812 User's Manual (Draft 0.5) available at **http://www.analog.com/microconverter**

14. UART FUNCTIONALITY - BAUD RATE GENERATION IN SPLIT TIMER MODE:

Background :   Timer 0 can be enabled as two 8-bit timers (Mode 3). Since the overflow flag for timer 1 is used by timer 0 in split timer mode, timer 1 can only be used as either a baud rate generator or as a timer as long as it does not require interrupts.

Issue :   When the timers are configured as above with timer 1 as a baud rate generator the reload value in TH1 fails to be loaded into TH0, generating the incorrect baud rate.

Work-Around :   Pending

Related Issues :   All other Timer and Baud Rate generation modes operate correctly.

REV. E.2   Sept/01

15. RESET FUNCTIONALITY WHEN EXECUTING FROM EXTERNAL MEMORY ONLY:

Background : The ADuC812 can be reset using in three ways, namely a. Power cycling the part, b. Toggling the reset pin and c. A watchdog Timeout

Issue :
a. When executing from external memory only i.e. $\overline{EA}$ tied low, resetting the ADuC812 by toggling the RESET pin can cause the part to enter an internal emulation mode.
b. When executing from external memory only i.e. $\overline{EA}$ tied low, a watchdog reset can cause the part to enter an internal emulation mode.

Work-Around :
a. A subsequent toggle of the reset pin will reset the part.

b. Tying the $\overline{PSEN}$ pin low through an external pull-down resistor (maximum value = 4KOhm) prevents entry to the emulation mode and correctly resets the ADuC812.

Related Issues :
a. This issue does not occur when executing code from **internal** memory only.
b. This issue does not occur when executing code from both **internal and external** code memory.
c. This issue does not occur when power cycling the ADuC812.